

## **AMENDMENTS TO THE CLAIMS:**

### **Complete Listing of Claims**

Claim 1. (currently amended) A clock recovery device comprising:

a number of sample components that obtain samples of a received NRZI encoded serial data stream at a number of phases, respectively, wherein the number of phases are successively offset throughout a bit time period and the samples are obtained throughout the time period at the number of phases;

a number of transition detectors corresponding to the number of obtained samples that analyze consecutive data samples in order to identify transitions;

a continuously operating blocking component that selectively blocks ~~one~~ clock ~~phases~~ phase which ~~were~~ was ~~possibly~~ selected for a prior transition ~~that would from erroneously contribute contributing to a~~ the final recovered clock if not blocked, while allowing clock phases which were selected for a prior transition and that do not erroneously contribute to the final recovered clock, and

a select clock component that selects a clock phase according to a ~~the~~ data toggle phase and generates a selected clock.

Claim 2. (original) The device of claim 1, wherein the sample components are comprised of D-type flip flops respectively clocked at one of the number of phases.

Claim 3. (original) The device of claim 1, wherein the number of obtained samples is 16.

Claim 4. (original) The device of claim 1, wherein the number of sample components are arranged in a dual column configuration to mitigate metastability.

Claim 5. (original) The device of claim 1, wherein the transition detectors comprise an XOR logic that generates a low value on a non-occurrence of a transition between two associated successive samples and generates a high value on an occurrence of a transition between two associated successive samples and a register that maintains the generated value.

Claim 6. (original) The device of claim 1, wherein the transition detectors generate data toggles that indicate occurrence or non-occurrence of a transition for respective time periods.

Claim 7. (currently amended) The device of claim 1, wherein the blocked ~~possibly-selected~~ clock phase is  $n+1$ , where  $n$  is a current phase with its data toggle set or high.

Claim 8. (currently amended) A clock and data recovery system comprising:  
a receiver that receives an NRZI encoded serial data stream having an associated frequency;

a phase generator that produces  $N$  phases of a clock signal that ~~has~~ have an associated frequency of approximately the same ~~about the~~ frequency of the received serial data stream, wherein the  $N$  phases are successively offset by  $(1/N)$  of a time period; and

a clock recovery component that identifies transitions in the received serial data stream, selects a clock phase from a first identified transition of a bit time, blocks one other stale selected clock phase, and selects an appropriate a clock accordingly.

Claim 9. (original) The system of claim 8, further comprising:

a data recovery component that identifies transitions in the received serial data stream and obtains a recovered serial data stream based solely on the identified transitions.

Claim 10. (currently amended) The system of claim 8, wherein the N phases are ~~substantially~~ evenly spaced.

Claim 11. (original) The system of claim 8, wherein N is equal to 8, 10, 11, 12 or 16 and the time period is equal to about 2.08333 ns.

Claim 12. (original) The system of claim 8, where the minimum value for N is set based on jitter, such that  $N > 2 / (1 - (2 * \text{jitter\_ratio}))$  wherein the jitter\_ratio is jitter rate expressed as a fraction.

Claim 13. (original) The system of claim 12, wherein the jitter is an estimated value.

Claim 14. (original) A clock recovery system comprising:

a number of phase components that respectively obtain data samples of an NRZI encoded serial data stream, identify transitions, set a toggle bit per transition, select a clock phase based on a prior phase toggle setting, use the preceding toggle to block the current clock phase, and generate a clock; and

a clock selector that generates a recovered clock from the clocks generated by the number of phase components.

Claim 15. (original) The system of claim 14, wherein the respective phase components use a data toggle to block a stale clock phase selection.

Claim 16. (currently amended) The system of claim 15, wherein jitter causes a ~~can cause~~ the toggle bit for a new transition to be set one phase before the now-stale clock phase selected for the prior transition.

Claim 17. (original) The system of claim 14, wherein the clock selector obtains the recovered clock by a logical OR of the respective clocks generated by the number of phase components.

Claim 18. (original) A method of recovering data over a single time period comprising:

- obtaining a number of data samples of a received NRZI encoded serial data stream according to a number of phase clocks;
- analyzing consecutive data samples to identify transitions;
- setting a toggle phase according to a first identified transition of a current bit time;
- blocking one subsequent phase from being selected as a clock; and
- selecting a clock phase according to the toggle phase.

Claim 19. (original) The method of claim 18, further comprising generating a number of phase clocks successively and evenly offset throughout a time period.

Claim 20. (original) The method of claim 18, wherein analyzing consecutive samples comprises performing an exclusive-or operation on the consecutive samples.

Claim 21. (original) The method of claim 18, wherein the subsequent phase to be blocked immediately follows the phase on which the toggle was set for the new transition.

Claim 22. (currently amended) The method of claim 18, wherein the clock phase is best selected to be  $N/2$  phases after the toggle phase, where  $N$  is the total number of data samples, ~~but can be some other number of phases after the toggle phase as long as the selected phase always falls between earliest and latest expected toggles.~~

Claim 23. (new) The method of claim 18, wherein the clock phase is selected to be a number of phases after the toggle phase such that the selected phase always falls between earliest and latest expected toggles.